AMENDMENTS TO THE SPECIFICATION

Please amend paragraphs [0046], [0047], [0048], and [0049] as follows:

[0046] Figure 7 shows a second embodiment of the code generator 200. Code generator 200 is substituted for code generator 30 and generates four-phase pseudo-random code sequences similar to those generated by the code generator 200 which greatly improve auto correlation properties and cross correlation properties. The second embodiment is an example of the second suboptimum implementation of Equation (11). Although four-phase sequences of any length can be generated, a length of 127 bits is selected as an example. Further, for the purposes of this example, there are N number of chips in a symbol, which represents the processing gain. A number M is selected to be relatively prime to N. The number of bits L required to provide a binary representation of processing gain N is determined by solving Equation (12). Since M=127 in this example, L=8. Therefore (M/N) is ~~sixteen~~ eight bits in length.

[0047] The code generator 30 includes an accumulator 210 which is L bits in length. Accumulator 210 has a length of 8 bits. A "1" is preferably applied to one input of accumulator 210. The number from flip flops 220_1 through 220_L is applied to a second input of the accumulator 210. Flip flops 220_1 through 220_L may be replaced by a shift register. Although bits are input to flip flops 220_1 through 220_L and accumulator 210 in parallel, the bits could be input in series. The sum of the two numbers input into the accumulator 210 is transmitted to flip flops 220_1 through 220_L. The output of flip flops 220_1 through 220_L are transmitted to flip flops 230_1 through 230_L as well as mixer 240. The mixer 240 also receives the output of flip flops 230_1 through 230_L. The accumulator 210 and flip flops 220_1-220_L, flip flops 230_1-230_L, and mixer 240 provide a flip flop feedback circuit. The

output of mixer 240 is input to mixer 250. Mixer 250 also receives an 8 bit input from (M/N). The extractor 260 extracts the ~~fifth and sixth~~ sixth and seventh least significant bits from the mixer 250. The ~~sixth~~ seventh least significant bit output from extractor 260 is converted to an I value by converter 280. The ~~fifth and sixth~~ sixth and seventh least significant bits are applied to an exclusive-or gate 270. The output of the exclusive-or gate 270 is converted to a Q value by a converter 290 as shown in Figure 8. The I and Q values output from converters 280 and 290 are applied to spreader 20 or despreader 140. As indicated before, (M/N) is an eight bit number in this example. Flip flops 220_1 through 220_l output the k value and flip flops 230_1 through 230_L output the k+1 value to the mixer 240. The mixer 250 receives the output of mixer 240 and the product of (M/N). When 2(M/N)k(k+1) is mapped to one of the four values {0, 1, 2, 3} by taking modulo 4, the result is the ~~fifth and sixth~~ sixth and seventh bits from extractor 260 (Figure 8).

[0048] Figure 9 is a flow diagram of the method performed by the circuit shown in Figure 7. The initial parameters M and N are loaded into registers or memory (not shown) before performing the dividing function (M/N). In addition, the value k is preferably equal to zero. The remaining apparatus in the second embodiment of the code generator 200 is also initialized (S1). The value of (M/N)k(k+1) is calculated (S2). The ~~fifth and~~ ~~sixth~~ sixth and seventh bits resulting from the above calculation are extracted (S3) in order to be converted into I and Q values (S4 and S5). The bits ~~(L-2) and (L-3)~~ (L-1) and (L-2) should be mapped to QPSK constellation as follows:

00→11

01→1-1

10→-1-1

11→-11

This mapping can be done in software or hardware by using first:

~~(L-2)~~ (L-1)	~~(L-3)~~ (L-2)		~~(L-2)~~ (L-1)	~~(L-2) ⊕ (L-3)~~ (L-1) ⊕ (L-2)
0	0	→	0	0
0	1	→	0	1
1	0	→	1	1
1	1	→	1	0

and then using the standard 0→1, 1→-1 mapping.

[0049] For example, if the ~~sixth~~ seventh bit for L-2 is equal to zero, then the I value is 1. If the ~~sixth~~ seventh bit is a 1, then the I value is -1. In the case of the Q value, if the output of the exclusive-or gate 270 is a zero, the Q value is 1. If the output of the exclusive-or gate 270 is a 1, the Q value is -1. The I and Q values are output to the spreader 20 or the despreader 140 (S6). The k value is incremented. Method steps S2 through S7 are repeated into all the digital data supplied by switch 14 is transmitted where all the data is received by switch 190.